

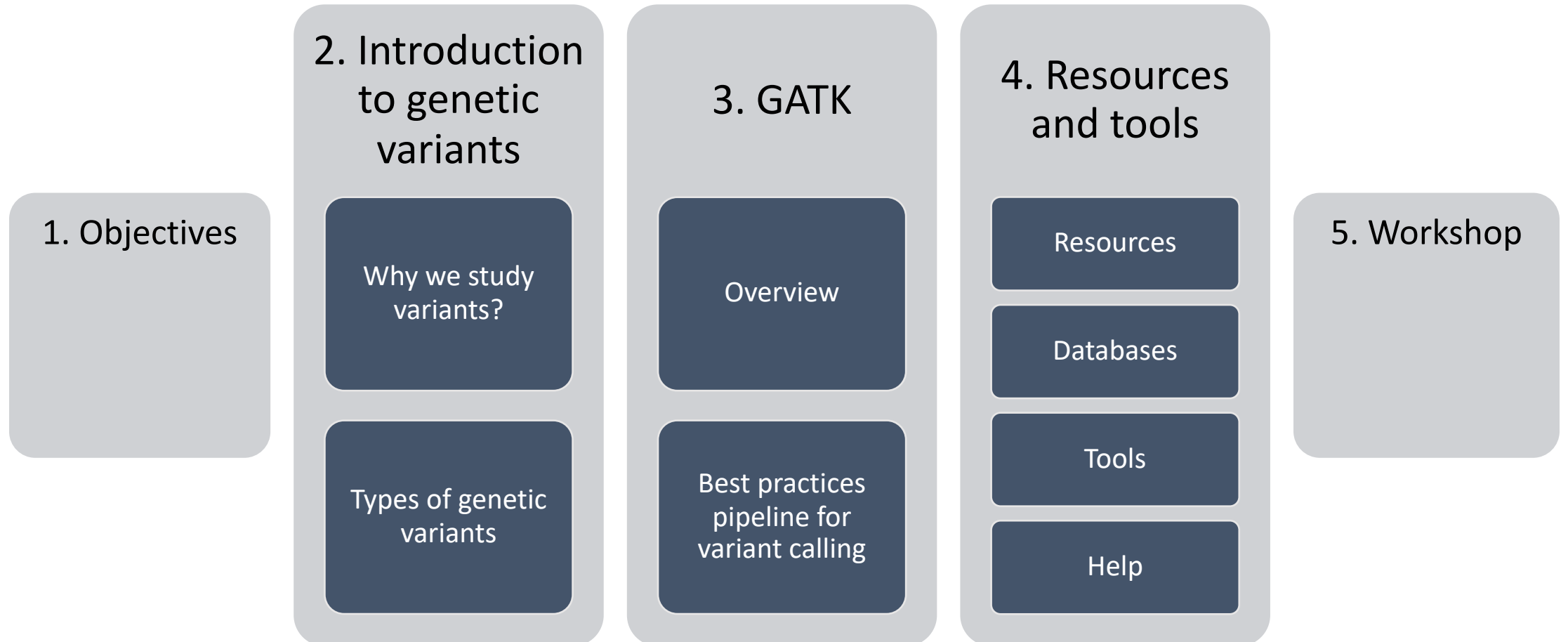
# Variant calling using GATK

Khalid Mahmood

2022

[https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant\\_calling\\_gatk1/variant\\_calling\\_gatk1/](https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant_calling_gatk1/variant_calling_gatk1/)

# Workshop overview



# 1. Objectives

- We aim to cover:
  - Perform QC of sequencing data
  - Align raw reads to reference sequences
  - Perform alignment metric and generating a QC report
  - Prepare alignment data for variant calling
  - Identify simple variants using GATK HaplotypeCaller
  - Visualise simple variant data (VCF files)
  - Perform basic variant filtering

## 2. Introduction to genetic variants

- There are approximately 3 billion base pairs in the human genome
- Humans share 99.5% of DNA with other humans
- A **variant** is a difference between similar genomes.
- Usually a difference between DNA sequences we are studying and a **reference genome**.
- To describe a variant we give the location (genomic coordinates) and genetic change.

*e.g.*      chr2                      9834      A→G

## 2. Introduction to genetic variants

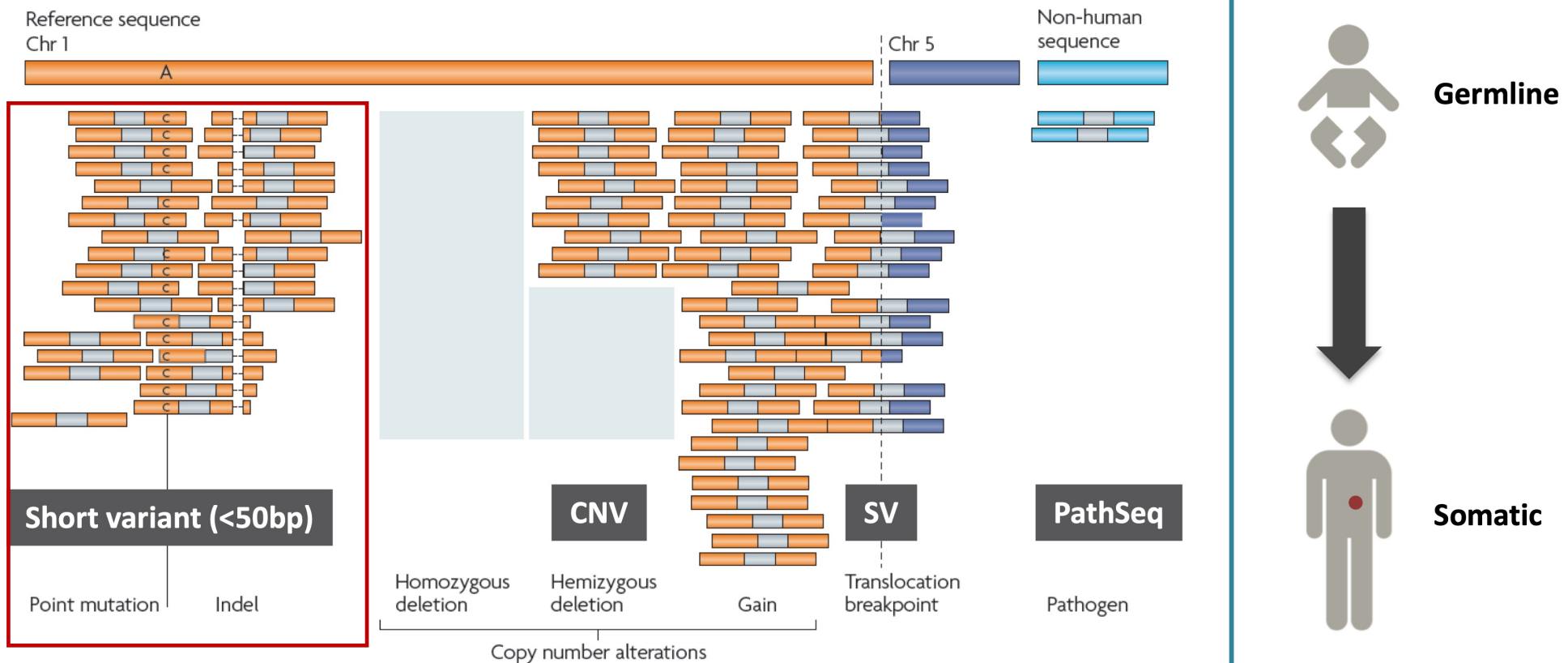
There is high degree of similarity but the human genome is large ~3 billion nucleotides.

This results in approximately 4-5 million variants between any individual and the reference genome.

These, seemingly small number of variations likely explains a significant proportion of phenotypic diversity among humans.

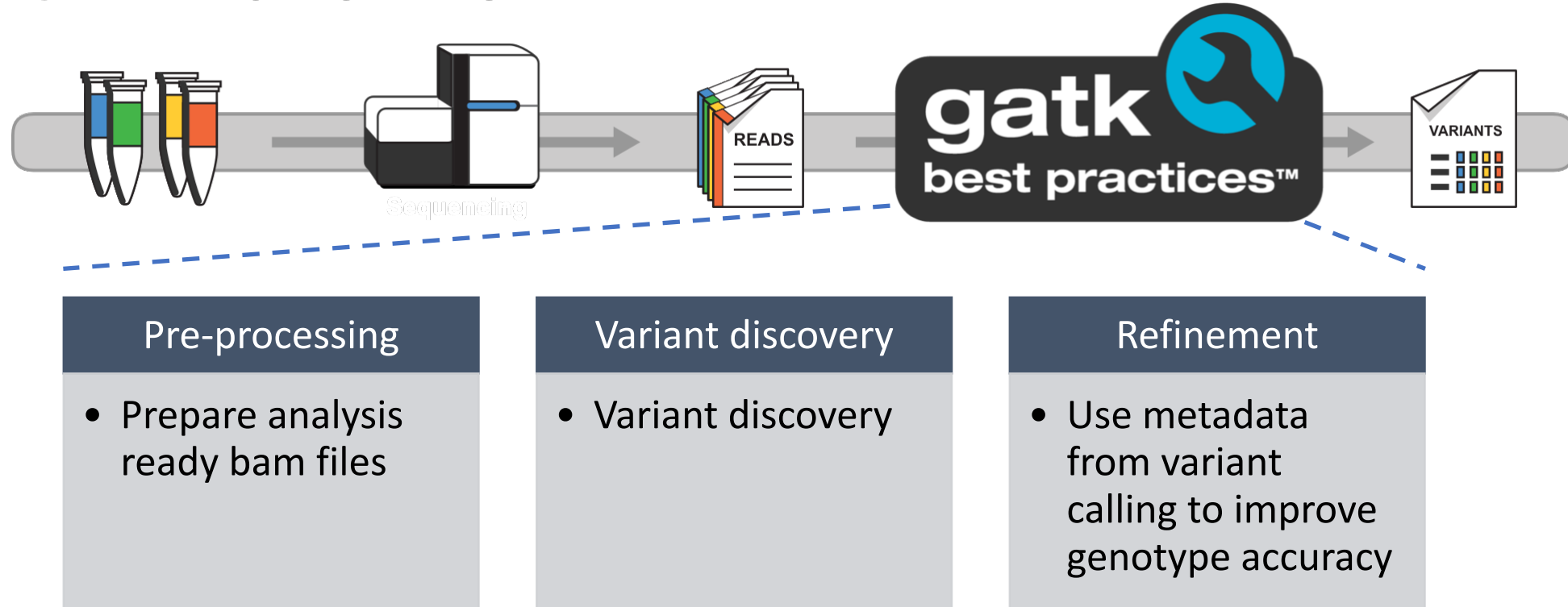
## 2. Types of genetic variants

Types of genetic variants:



**Focus of this workshop:**  
**Calling short germline variants**

### 3. GATK overview



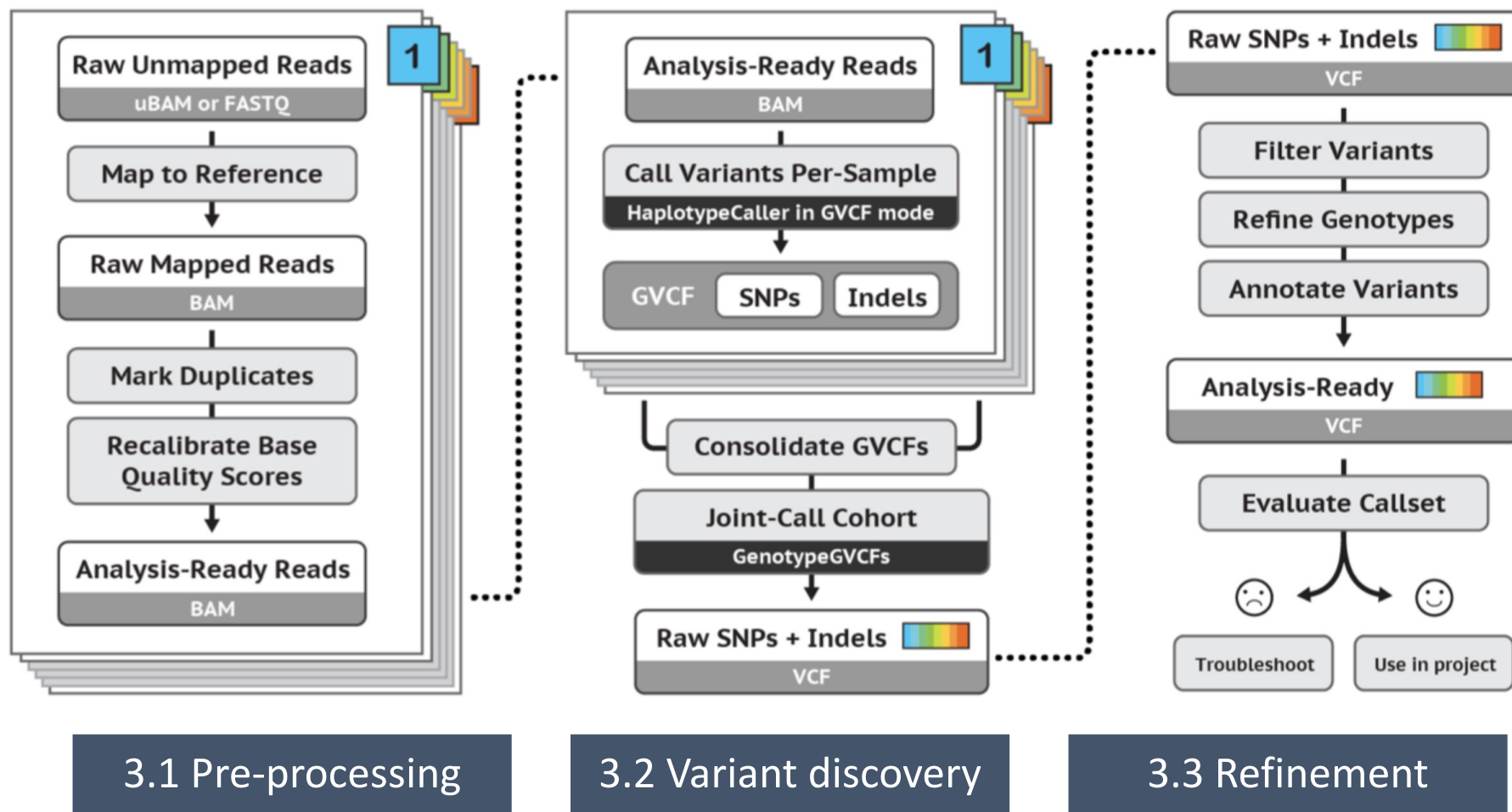
- Genome Analysis Toolkit (GATK): software package to analyze high-throughput sequencing data

## 3. GATK overview

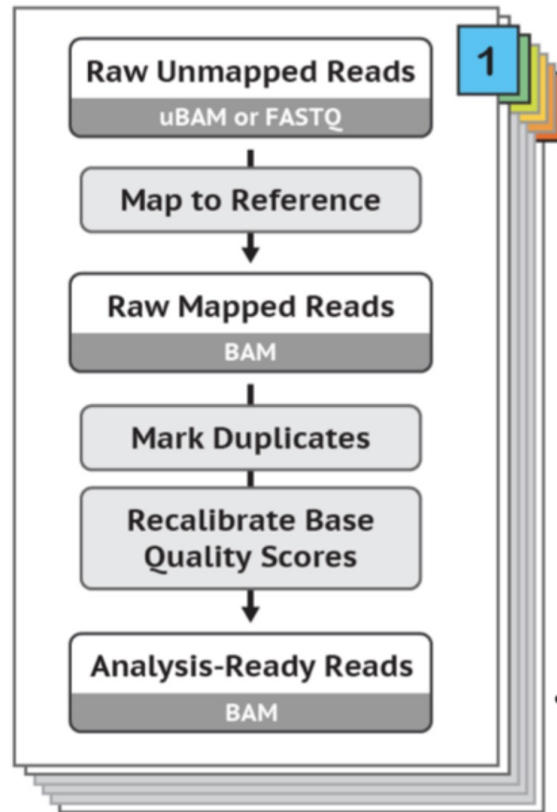
- Download available from
  - <https://github.com/broadinstitute/gatk/releases>
  - Tutorial version: GATK 4.2.0.0 (September 2021)
  - Current version: GATK 4.2.2.0
- Explore GATK website - [gatk.broadinstitute.org](http://gatk.broadinstitute.org)
  - Tool index – provides tools usage instructions
  - Technical documentation – provides details on for example Algorithms
  - Forum – provides access to Q&As and community discussions



# 3. GATK Best practices pipeline



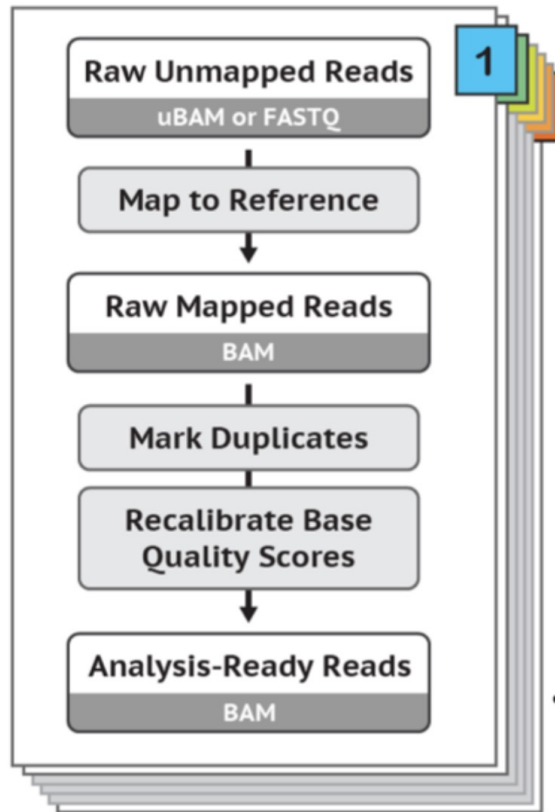
## 3.1 Pre-processing



Pre-processing

- A sequencing experiment results in a large volume of sequencing reads
- Reads are not mapped to a reference
- Reads can contain errors and technical artifacts
- e.g. a molecule sequenced multiple times will result in duplicate reads
- We need to filter and prepare the reads and the alignment data – ready for variant calling

## 3.1 Map to reference

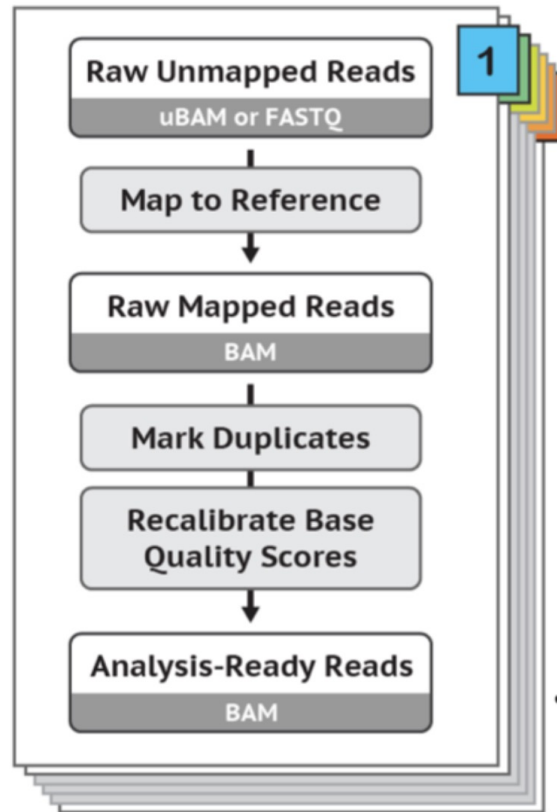


Pre-processing

- **BWA-MEM**

- `bwa mem -M -t 4 -R "@RG\tID:SRR622461.7\tSM:NA12878\tLB:ERR194147\tPL:ILLUMINA" <reference> sample_1.fastq sample_2.fastq > alignment.sam`
- -M: inserts a tag to the alignment if non-primary alignment (required by GATK)
- -R: read group
- -t: threads or number of cpus
- <reference>: path to reference genome in fasta format and the BWA index files

# 3.1 Map to reference



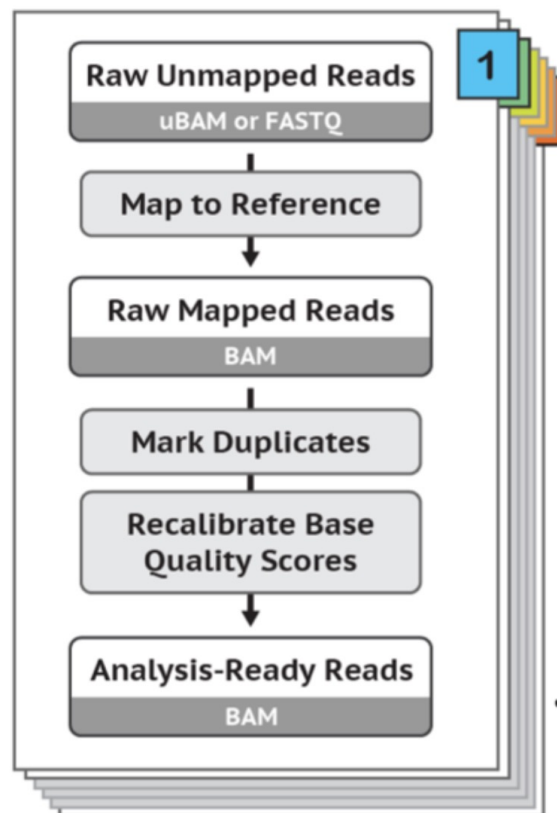
Pre-processing

## • BWA-MEM

- `bwa mem -M -t 4 -R "@RG\tID:SRR622461.7\tSM:NA12878\tLB:ERR194147\tPL:ILLUMINA"<reference> sample_1.fastq sample_2.fastq > alignment.sam`
- -R: read group contains information such as the sample name, library and flow cell.
- Refers to a set of reads generated from a single sequencing run in particular machine

|     |                |            |              |             |
|-----|----------------|------------|--------------|-------------|
| @RG | ID:SRR622461.7 | SM:NA12878 | LB:ERR194147 | PL:ILLUMINA |
|-----|----------------|------------|--------------|-------------|

# 3.1 Map to reference



Pre-processing

- Output is a SAM/BAM file.
- SAM file specifications:  
<https://samtools.github.io/hts-specs/SAMv1.pdf>

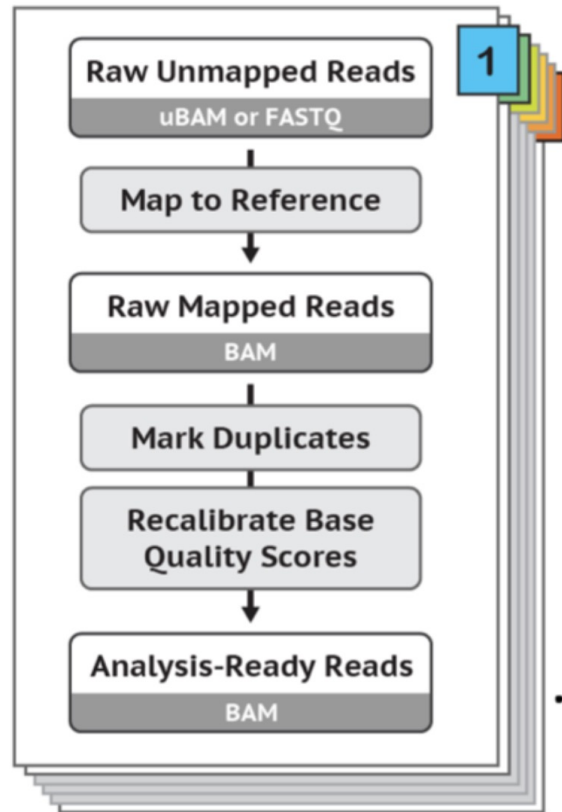
## Header

```
@HD VN:1.5 SO:coordinate
@RG ID:SRR622461.7 SM:NA12878 LB:ERR194147 PL:ILLUMINA
@PG ID:bwa PN:bwa VN:0.7.17-r1188 CL:bwa mem -M -t 4 -R
```

## Alignment

| read name    | flag | position                            | CIGAR            | read                   | flags/<br>metadata |
|--------------|------|-------------------------------------|------------------|------------------------|--------------------|
| ERR194147.45 | 163  | chr18 6576006 99 101M = 6578028 317 |                  | CATTCT... <B<<BBBBB... | NM:i:0 MD:Z:101... |
|              |      | mapping quality                     | mate information |                        | PHRED quality      |

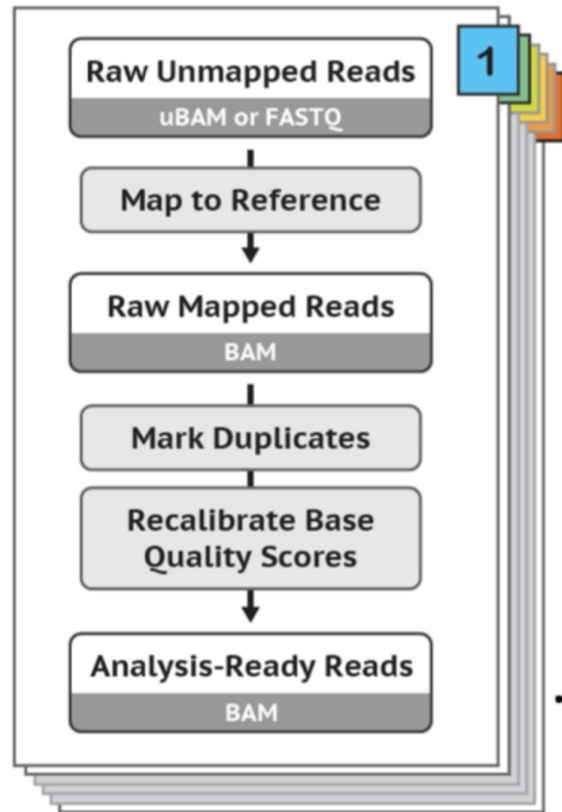
## 3.1 Mark duplicates



Pre-processing

- **Mark Duplicates**
- Identify reads that are non-independent measurement of sequence fragment
  - Same template of DNA sampled multiple times
  - PCR duplicates
- High sequence identify
- Align to same reference position

## 3.1 Mark duplicates

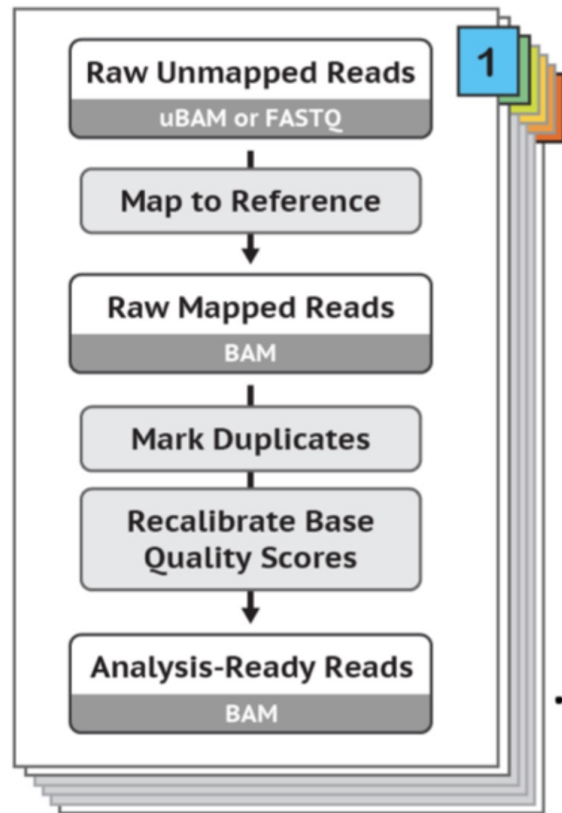


Pre-processing

- **Mark Duplicates**

- `gatk MarkDuplicates -I sample.bam -O sample.dedup.bam -M sample.dedup.metrics.txt`
- Recommended to be performed on reads per library or lane
- SAM flags are used to mark reads as duplicates
- Downstream GATK tools depend on these flags to assess support for variants and alleles

## 3.1 Base recalibration



Pre-processing

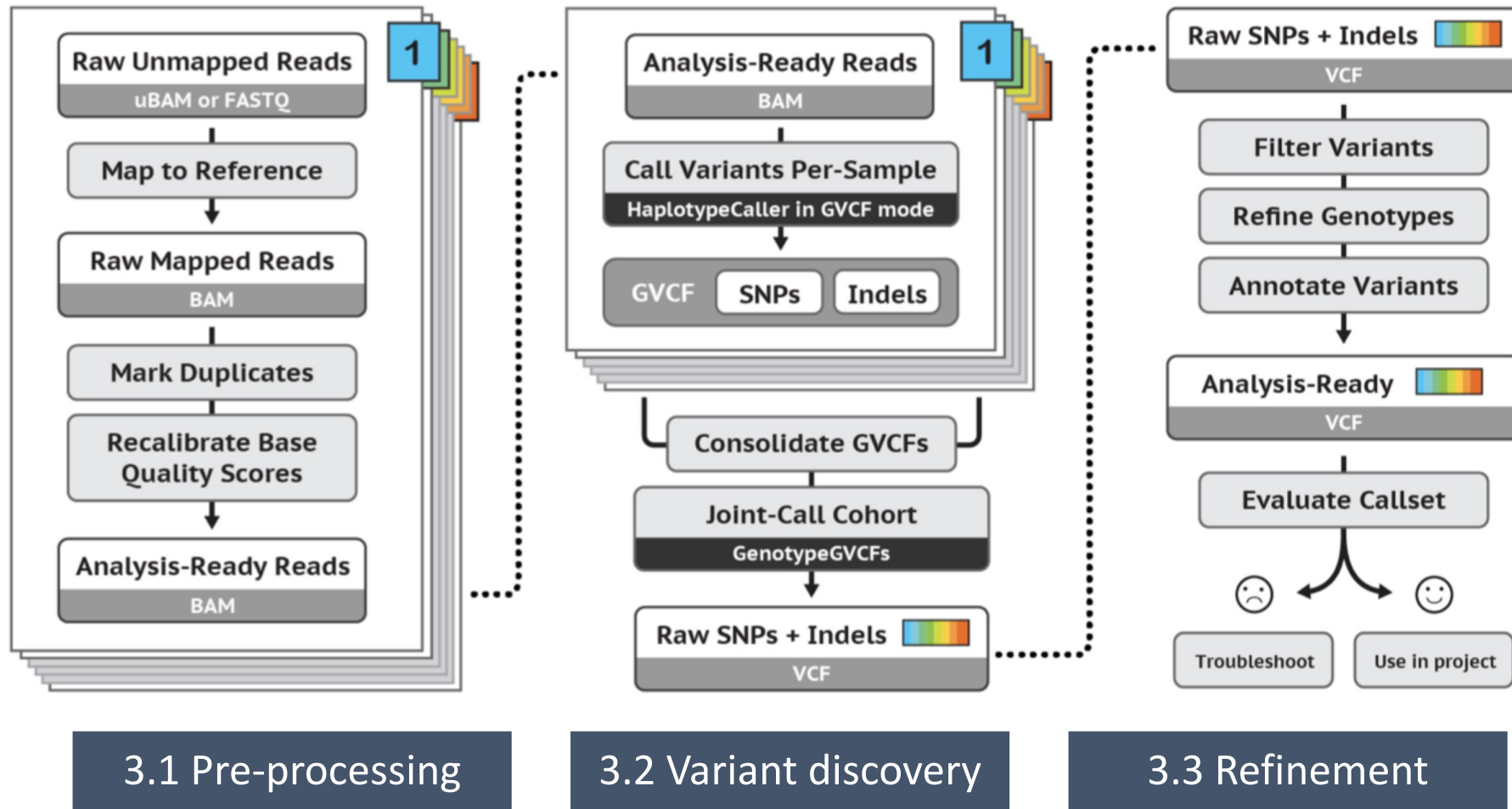
- **Base recalibration**

- **gatk tools BaseRecalibrator and ApplyRecalibration**

- Performed per-sample to detect and correct for patterns of systematic errors in base quality scores.
- Evidenced by calculating metrics based on known variant locations
- Important for building reliable evidence for downstream analysis.

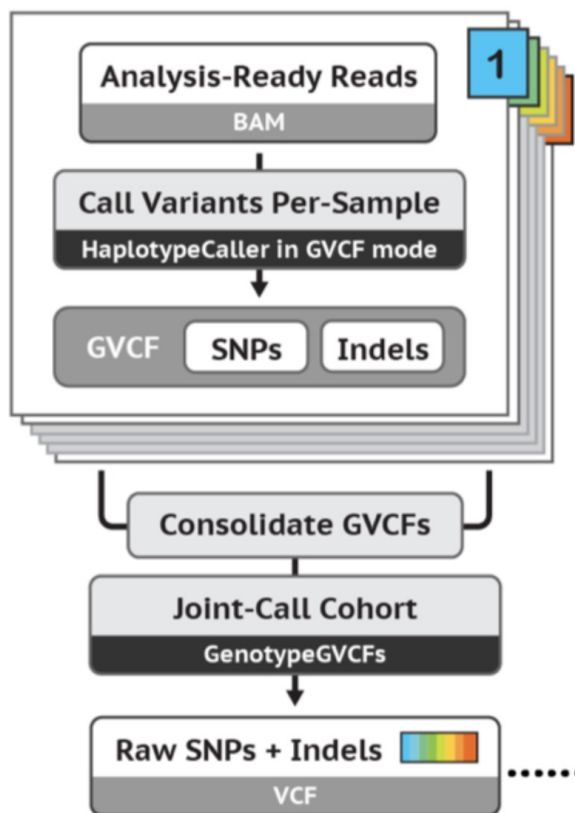


### 3. GATK Best practices pipeline

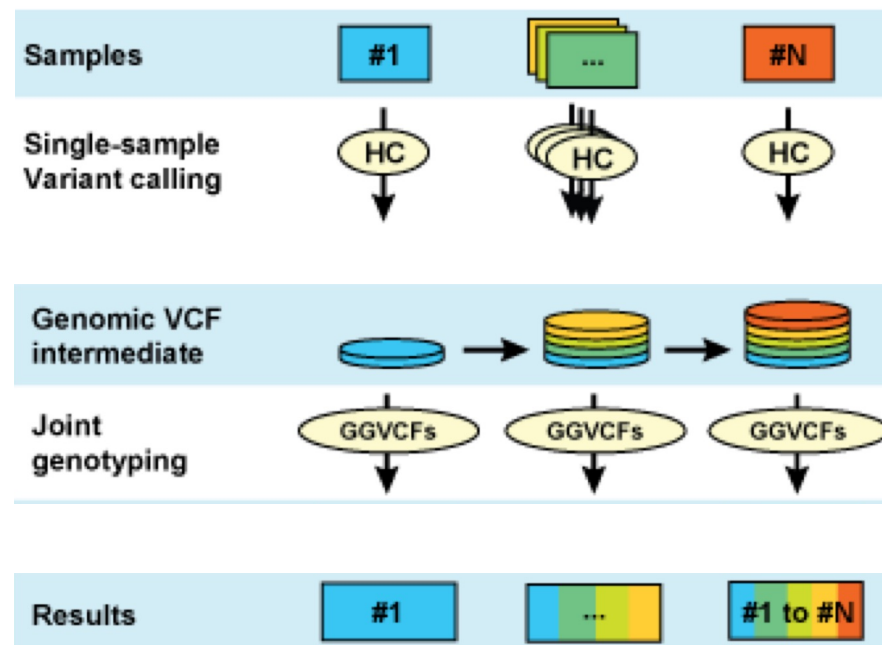


## 3.2 Variant discovery

### • Software



Variant discovery



HaplotypeCaller

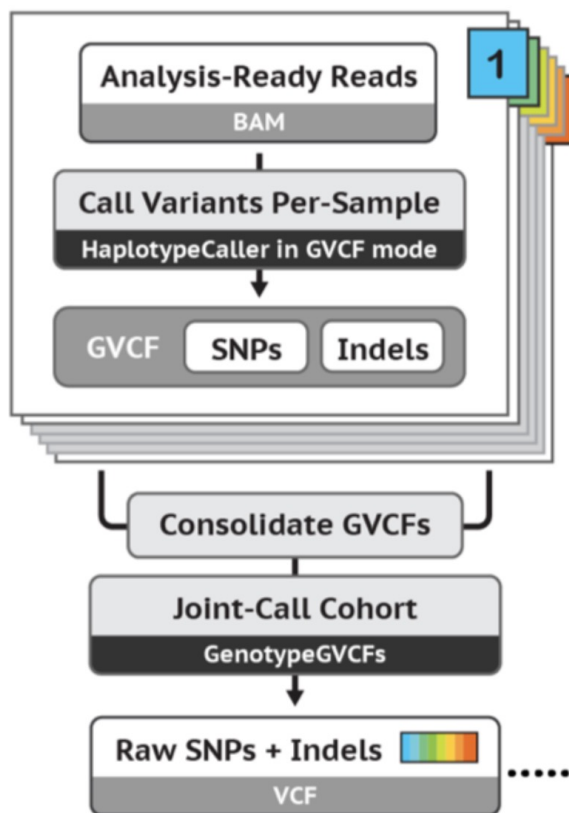
CombineGVCfs/  
GenomicsDBImport

GenotypeGVCfs

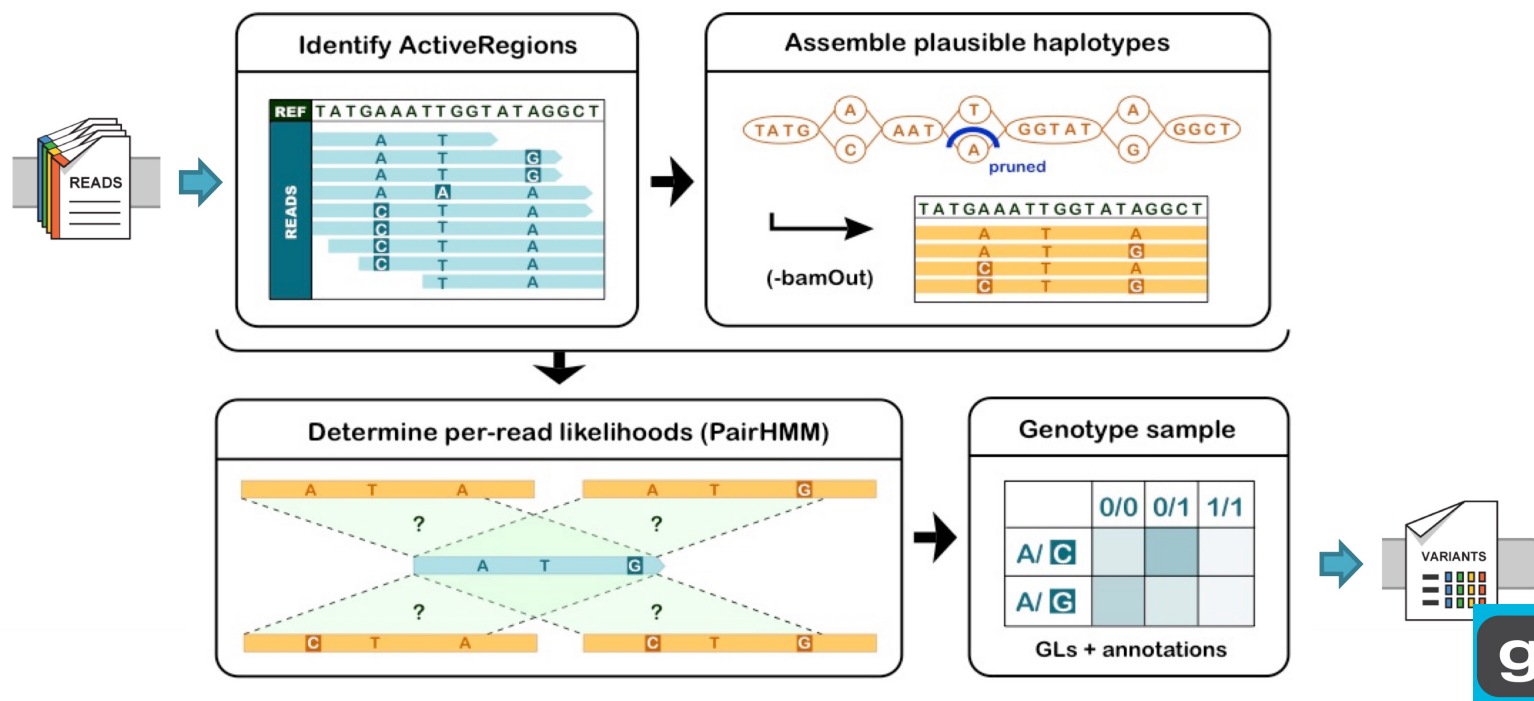
## 3.2 Variant discovery

### • HaplotypeCaller

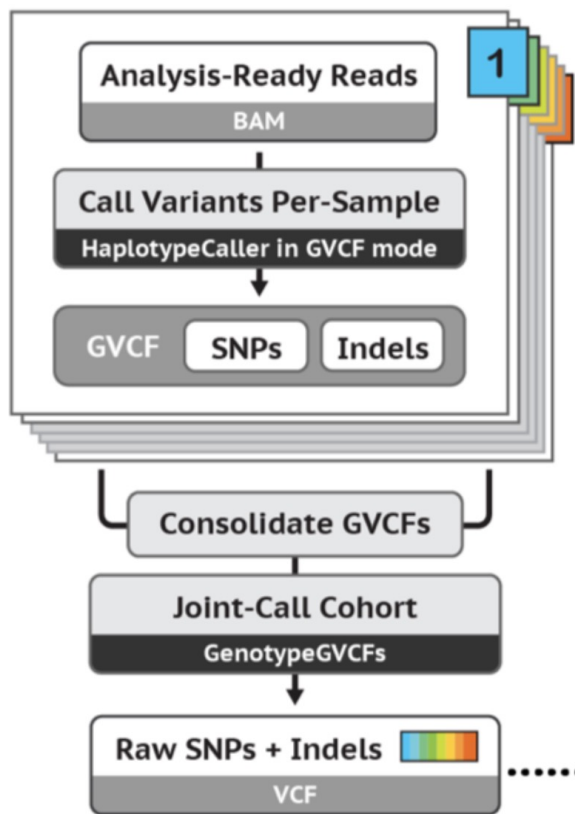
- `gatk --java-options "-Xmx4g" HaplotypeCaller -R <reference.fa> -I input.bam -O output.g.vcf.gz -ERC GVCF`



Variant discovery



## 3.2 Variant discovery



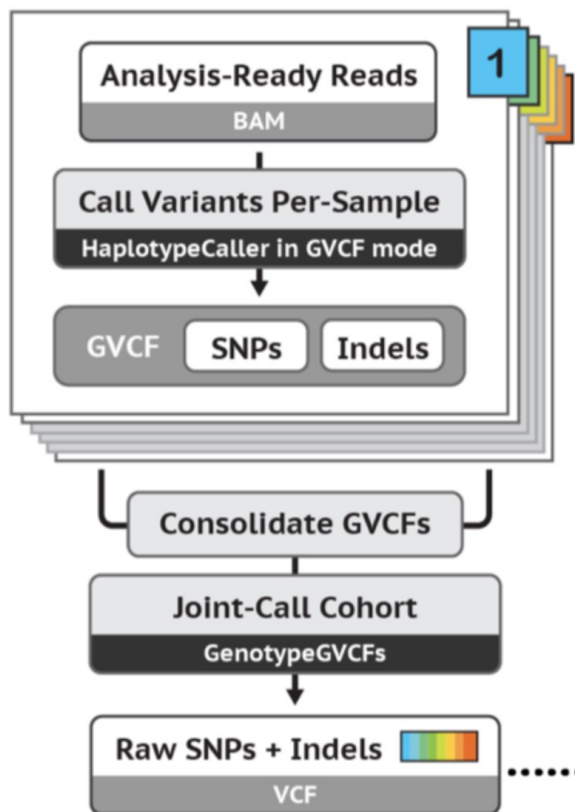
- **CombineGVCfs**

- `gatk CombineGVCfs R <reference.fa> --variant sample1.g.vcf.gz --variant sample2.g.vcf.gz -O cohort.g.vcf.gz`

- Combine per samples gVCF files (produced by HaplotypeCaller) into a multi-sample gVCF file.

Variant discovery

## 3.2 Variant discovery



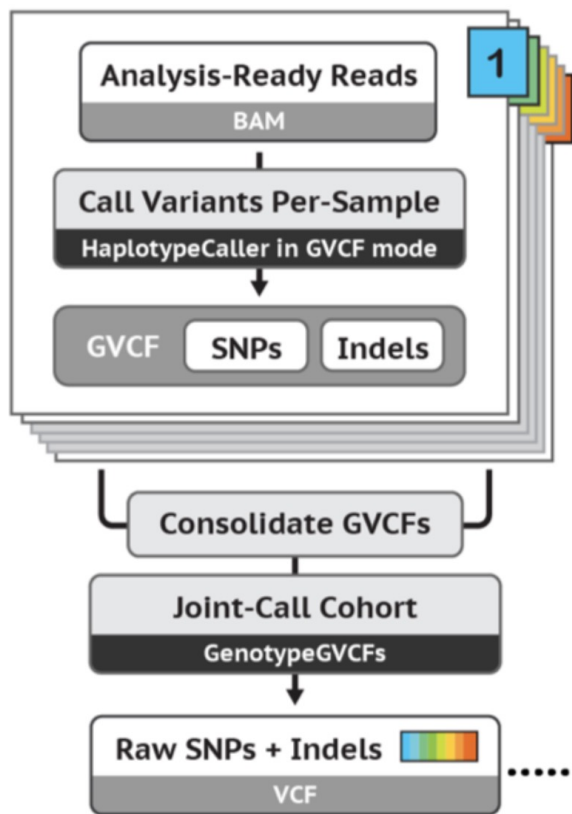
- **GenotypeGVCfs**

- `gatk --java-options "-Xmx4g" GenotypeGVCfs -R <reference.fa> -V cohort.g.vcf.gz -O output.vcf.gz`

- Combine per samples gVCF files (produced by HaplotypeCaller) into a multi-sample gVCF file.

Variant discovery

## 3.2 Variant discovery



- Output is a VCF file
- VCF file specifications

<https://samtools.github.io/hts-specs/VCFv4.2.pdf>

### Header

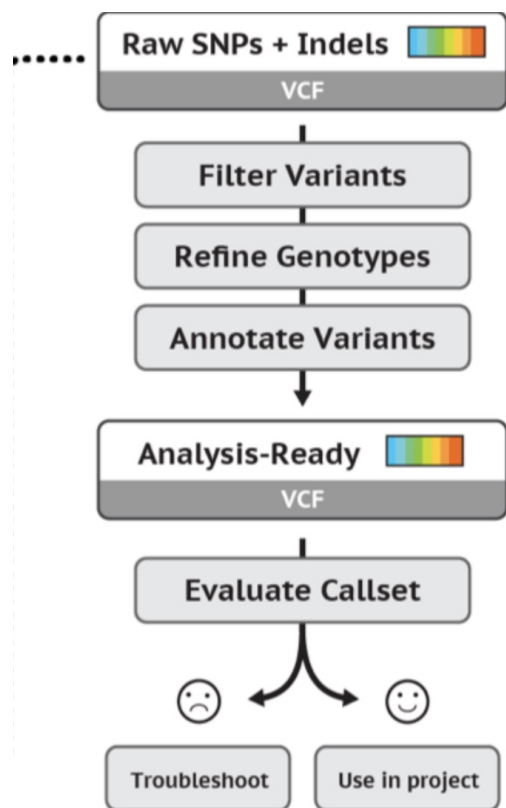
```
#fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##contig=<ID=1,length=249250621>
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
```

### Variant record

| #CHROM | POS    | ID | REF | ALT | QUAL  | FILTER | INFO       | FORMAT   | Sample1      |
|--------|--------|----|-----|-----|-------|--------|------------|----------|--------------|
| 1      | 567376 | .  | G   | A   | 146.3 | PASS   | AC=1;DP=55 | GT:AD:DP | 0/1:30,25:55 |

Variant discovery

## 3.3 Variant Refinement



- Variant callers are sensitive
- The aim here is to identify potential false positives and apply filters to remove those less likely to be real variants. Strategies include:
  1. Variant quality score recalibration (using known sites)
  2. Hard filtering on quality criteria
  3. Annotation features

### Variant record

| #CHROM | POS    | ID | REF | ALT | QUAL  | FILTER | INFO       | FORMAT   | Sample       |
|--------|--------|----|-----|-----|-------|--------|------------|----------|--------------|
| 1      | 567376 | .  | G   | A   | 146.3 | PASS   | AC=1;DP=55 | GT:AD:DP | 0/1:30,25:55 |

## 4. Resources and tools

- GATK resources bundle: collection of files for GATK based analysis working with human sequencing data.
- <ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/hg38>

1000G\_omni2.5.hg38.vcf.gz

1000G\_phase1.snps.high\_confidence.hg38.vcf.gz

Axiom\_Exome\_Plus.genotypes.all\_populations.poly.hg38.vcf.gz

dbsnp\_146.hg38.vcf.gz

hapmap\_3.3\_grch38\_pop\_stratified\_af.vcf.gz

hapmap\_3.3.hg38.vcf.gz

Homo\_sapiens\_assembly38.dict

Homo\_sapiens\_assembly38.fasta

Homo\_sapiens\_assembly38.fasta.gz

Mills\_and\_1000G\_gold\_standard.indels.hg38.vcf.gz



## 4. Resources and tools

- BWA-MEM index
- `bwa index Homo_sapiens_assembly38.fasta`

Homo\_sapiens\_assembly38.fasta

Homo\_sapiens\_assembly38.fasta.amb

Homo\_sapiens\_assembly38.fasta.ann

Homo\_sapiens\_assembly38.fasta.bwt

Homo\_sapiens\_assembly38.fasta.pac

Homo\_sapiens\_assembly38.fasta.sa

## 4. Resources and tools

| Tools name | function  |
|------------|---|
| FastQC     | QC tools for raw sequencing reads   |
| MultiQC    | QC report aggregator (generates an HTML report)                                   |
| GATK       | Set of tools for variant calling  |
| Picard     | A command line tool to analysis and manipulate sequencing files                   |
| Samtools   | Suite of tools for interacting with mapped sequencing reads (SAM/BAM/CRAM format) |
| BCFtools   | Suite of tools for interacting with variant data (VCF/BCF formats)                |

---

## 4. Help

- Tool documentation
- GATK forum
- Online resources (e.g. Biostar)
- GitHub for technical issues/discussions

# Workshop structure

# Workshop structure

Tutorial content

Bioinformatics Documentation

Home

Tools and skill development >

Genomics >

Variant detection >

Introduction to Variant detection

Variant Calling part 1 (Galaxy)

Variant Calling part 2 (Galaxy)

Long-read Structural Variant Calling

Variant calling using command-line tools

de novo assembly >



Metabarcoding >

Transcriptomics >

Proteomics >

Statistics and Visualisation >

Structural Modelling >

## Variant calling using GATK4

Anticipated workshop duration when delivered to a group of participants is **4 hours**.

For queries relating to this workshop, contact Melbourne Bioinformatics ([bioinformatics-training@unimelb.edu.au](mailto:bioinformatics-training@unimelb.edu.au)).

### Author Information

Khalid Mahmood  
Melbourne Bioinformatics, University of Melbourne  
Developed: July 2021  
Reviewed: August 2021

### Overview

Topic

### Table of contents

Author Information

Overview

Learning Objectives

Description

Requirements and preparation

Mode of Delivery

Byobu-screen

Tutorial setting

The Genome Analysis Toolkit (GATK)

How this tutorial works

Tutorial contents table

Section 1: Map raw mapped reads to reference genome

1. Preparation and data import
2. Align genome

Section 2: Prepare analysis ready reads

1. Sort SAM/BAM
2. Mark duplicate reads
3. Base quality recalibration

Section 3: Variant calling

1. Apply HaplotypeCaller
2. Apply CombineGVCFs
3. Apply GenotypeGVCFs

Section 4: Filter and prepare

Tutorial navigation

# Workshop structure

Command and output blocks  
‘#’ comments - do not run

- Tools and skill development >
- Genomics >
  - Variant detection >
    - Introduction to Variant detection
    - Variant Calling part 1 (Galaxy)
    - Variant Calling part 2 (Galaxy)
    - Long-read Structural Variant Calling
    - Variant calling using command-line tools
  - de novo assembly >
  - Metabarcoding >
  - Transcriptomics >
  - Proteomics >
  - Statistics and Visualisation >
  - Structural Modelling >

## 2. Apply CombineGVCFs

The CombineGVCFs tool is applied to combine multiple single sample VCF files, merging them into a single multi-sample VCF file.

We have pre-processed two additional samples (NA12891 and NA12892) up to the HaplotypeCaller step (above). Let's first copy the VCF files to the output directory.

```
#let's make sure that we are in the appropriate directory
cd
cp /mnt/shared_data/NA12891.g.vcf.gz* output/.
cp /mnt/shared_data/NA12892.g.vcf.gz* output/.
```

```
gatk --java-options "-Xmx7g" CombineGVCFs \
-R reference/hg38/Homo_sapiens_assembly38.fasta \
-V output/NA12878.g.vcf.gz \
-V output/NA12891.g.vcf.gz \
-V output/NA12892.g.vcf.gz \
-L chr20 \
-O output/cohort.g.vcf.gz
```

### Let's look at the combined VCF file

```
less output/cohort.g.vcf.gz
```

Work your way down to the variant records? How many samples do you see in the VCF file? Hint: look at the header row.

Now that we have a merged VCF file, we are ready to perform genotyping.

## 3. Apply GenotypeGVCFs

GenotypeGVCFs

```
gatk --java-options "-Xmx7g" GenotypeGVCFs \
-R reference/hg38/Homo_sapiens_assembly38.fasta \
-V output/cohort.g.vcf.gz \
-L chr20 \
-O output/output.vcf.gz
```

Information

Visualisations: VCF file

- Overview
- Learning Objectives
- Description
- Requirements and preparation
- Mode of Delivery
- Byobu-screen
- Tutorial setting
- The Genome Analysis Toolkit (GATK)
- How this tutorial works
- Tutorial contents table
- Section 1: Map raw mapped reads to reference genome
  - 1. Preparation and data import
  - 2. Align genome
- Section 2: Prepare analysis ready reads
  - 1. Sort SAM/BAM
  - 2. Mark duplicate reads
  - 3. Base quality recalibration
- Section 3: Variant calling
  - 1. Apply HaplotypeCaller
  - 2. Apply CombineGVCFs
  - 3. Apply GenotypeGVCFs
- Section 4: Filter and prepare analysis ready variants
  - 1. Variant Quality Score Recalibration
  - 2. Additional filtering
  - 3. Final analysis ready VCF files
- Section 5: Exporting variant data and visualisation
  - 1. VariantsToTable
  - 2. HTML report

Interactive sections  
Notes, hints, exercises

# Workshop structure

## Table of contents

Author Information  
Overview  
Learning Objectives  
Description  
Requirements and preparation  
    Mode of Delivery  
    Byobu-screen  
Tutorial setting  
    The Genome Analysis Toolkit (GATK)  
    How this tutorial works

Introductory material  
Tutorial delivery and some instructions

Tutorial contents table  
Section 1: Map raw mapped reads to reference genome  
    1. Preparation and data import  
    2. Align genome  
Section 2: Prepare analysis ready reads  
    1. Sort SAM/BAM  
    2. Mark duplicate reads  
    3. Base quality recalibration  
Section 3: Variant calling  
    1. Apply HaplotypeCaller  
    2. Apply CombineGVCFs  
    3. Apply GenotypeGVCFs  
Section 4: Filter and prepare analysis ready variants  
    1. Variant Quality Score Recalibration  
    2. Additional filtering  
    3. Final analysis ready VCF file  
Section 5: Exporting variant data and visualisation  
    1. VariantsToTable  
    2. HTML report

## Workshop content:

- 5 sections
- Each section covers a stage in the variant calling pipeline
- Each section has a text explain the process and links to relevant material
- Sections have multiple steps. Mostly have an input and an expected output file.
- This is a pipeline: input to a step is the output from a previous step

# Workshop computers

- We will be conducting the workshop on virtual machines
- Hosted on the University of Melbourne Research Cloud service and the ARDC Nectar Research Cloud infrastructure.
- Infrastructure for development and setup of the workshops machines by Simon Gladman



# Workshop computers

- Each participant should have a username and a password
- Each participant will be assigned one of the following two VM machines:
  - 115.146.84.252
  - 115.146.84.226
- Configuration:

- Open a terminal window and on the command prompt type and enter:

```
khalidm~$ ssh alpha@115.146.84.252
alpha@115.146.84.252's password:
```

```
alpha@test-i2: ~  
alpha@test-i2: ~ (ssh)  
khalidm~$ ssh alpha@115.146.84.252  
alpha@115.146.84.252's password:  
-----  
Nectar Ubuntu 20.04.3 LTS (Focal Fossa)  
  
Image details and information is available at:  
  https://support.ehelp.edu.au/support/solutions/articles/6000106269  
-----  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
Last login: Tue Sep 14 08:16:18 2021 from 124.188.77.12  
(gatk4) alpha@test-i2:~$ █
```

```
alpha@test-i2: ~  
alpha@test-i2: ~ (ssh)  
  
1 [ 0.0%] 5 [ 0.0%] 9 [ 0.7%] 13 [ 0.0%]  
2 [ 0.0%] 6 [ 0.0%] 10 [ 0.0%] 14 [ 0.7%]  
3 [ 0.0%] 7 [ 0.0%] 11 [ 0.0%] 15 [ 0.0%]  
4 [ 0.0%] 8 [ 0.0%] 12 [ 0.0%] 16 [ 0.0%]  
Mem [|||||] 242M/31.4G Tasks: 29, 14 thr; 1 running  
Swp [ ] 1.01M/92.9M Load average: 0.01 0.03 0.00  
Uptime: 1 day, 01:16:07  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
40563 alpha 20 0 10516 3920 3276 R 0.0 0.0 0:00.10 htop  
706 root 20 0 389M 21892 10164 S 0.0 0.1 0:27.49 /usr/bin/python3 /usr/bin/fail2ban-server -xf sta  
705 root 20 0 389M 21892 10164 S 0.0 0.1 0:22.67 /usr/bin/python3 /usr/bin/fail2ban-server -xf sta  
644 root 20 0 389M 21892 10164 S 0.0 0.1 0:55.68 /usr/bin/python3 /usr/bin/fail2ban-server -xf sta  
1 root 20 0 164M 12368 8372 S 0.0 0.0 0:08.42 /lib/systemd/systemd --system --deserialize 27  
651 root 20 0 232M 7724 6500 S 0.0 0.0 0:02.81 /usr/lib/accountsservice/accounts-daemon  
677 root 20 0 232M 7724 6500 S 0.0 0.0 0:00.05 /usr/lib/accountsservice/accounts-daemon  
622 root 20 0 232M 7724 6500 S 0.0 0.0 0:03.25 /usr/lib/accountsservice/accounts-daemon  
624 root 20 0 9412 3020 2760 S 0.0 0.0 0:00.18 /usr/sbin/cron -f  
625 messagebu 20 0 7760 4432 3552 S 0.0 0.0 0:01.15 /usr/bin/dbus-daemon --system --address=systemd:  
638 root 20 0 81928 3644 3268 S 0.0 0.0 0:00.00 /usr/sbin/irqbalance --foreground  
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice +F8Nice +F9Kill F10Quit
```

# Useful Linux commands

- Autofill on command line: **Tab key**
- Abort command: **Ctrl-c**
- List contents of a directory: **ls -l**
- What's the path to my current directory: **pwd**
- Change directory: **cd <path/to/destination>**
- Create a directory: **mkdir <directory name>**
- Copy a file: **cp <source file> <destination path/name>**
- Remove a directory: **rmdir <directory name>**
- Remove a file: **rm <file name>**
- Rename/move a file (this is not copying a file): **mv <source file> <destination file>**
- Open a text editor: **nano**
- Print file content (small files): **cat <file name>**
- Print file content (quick view): **less <file name>**
- Print file content (quick view/first 10 lines of a file): **head <file name>**
- Print file content (quick view/last 10 lines of a file): **tail <file name>**
- curl or wget: download a file from a URL (you will see this in other QIIME2 tutorials)
- Documentation for a command line tool: try **man <tool name>** OR **<command\_name> --help**

# Workshop data

- Primary data: paired-end sequencing reads from the chr20
  - chr20:2677705-6631126
- Whole genome sequencing data
  - Female
  - Utah resident (European ancestry)
  - 1000 genomes project (NA12878)
- Other data from
  - A male and female
  - Utah resident (European ancestry)
  - 1000 genomes project (NA12891 and NA12892)

Byrska-Bishop, Marta et al. "High coverage whole genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios". *bioRxiv*. (2021).

# Byobu-screen

- A terminal multiplexer or a tool to to create multiple ‘windows’ in a single screen
- Improves stability of terminal sessions when connected to a remote computer
- List screen sessions: `byobu-screen -ls`
- Start new session: `byobu-screen -S workshop`
- Detach from screen to original window: `Ctrl-a-d`
- More details:
- [https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant\\_calling\\_gatk1/variant\\_calling\\_gatk1/#byobu-screen](https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant_calling_gatk1/variant_calling_gatk1/#byobu-screen)

# Workshop

[https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant\\_calling\\_gatk1/variant\\_calling\\_gatk1/](https://www.melbournebioinformatics.org.au/tutorials/tutorials/variant_calling_gatk1/variant_calling_gatk1/)

